

Convolutional Neural Networks for Sentence Classification

Yoon Kim, 2014

발표자 : 이은주



arXiv

<https://arxiv.org> > cs



Convolutional Neural Networks for Sentence Classification

Y Kim 저술 · 2014 · 16830회 인용 — Authors: **Yoon Kim**. Download a PDF of the paper titled **Convolutional Neural Networks for Sentence Classification**, by **Yoon Kim**. Download PDF.

- YOON KIM은 하버드 대학교에서 컴퓨터 과학 박사 학위를 취득 후 현재 MIT 조교수로 재직 중
- CNN을 사용한 NLP의 성능을 입증하여 주목을 받은 논문

Abstract

We report on a series of experiments with convolutional neural networks (CNN) trained on top of pre-trained word vectors for sentence-level classification tasks. We show that a simple CNN with little hyperparameter tuning and static vectors achieves excellent results on multiple benchmarks. Learning task-specific vectors through fine-tuning offers further gains in performance. We additionally propose a simple modification to the architecture to allow for the use of both task-specific and static vectors. The CNN models discussed herein improve upon the state of the art on 4 out of 7 tasks, which include sentiment analysis and question classification.

- 문장 수준의 분류를 위해 pre-trained 단어 벡터와 CNN을 사용한다.
- Fine-tuning을 통해 특정 task의 벡터를 학습하는 것은 성능 향상 시킬 수 있다.
- 4개의 간단한 CNN 모델을 사용하였다.
- 감정 분석과 질문 분류를 포함한 7개의 데이터셋 중 4개에서 최고의 성능을 보여주었다.

Sentence Classification Task

: 감정 분류

ex) 이번 논문은 대박이야! → 긍정

ex) 이번 음악 너무 별로인데? → 부정

: 주제 분류

ex) T1, 전승으로 1황 '등극' → E-스포츠

ex) 엘지에너지솔루션 따상 가나? → 경제

: 질문 분류

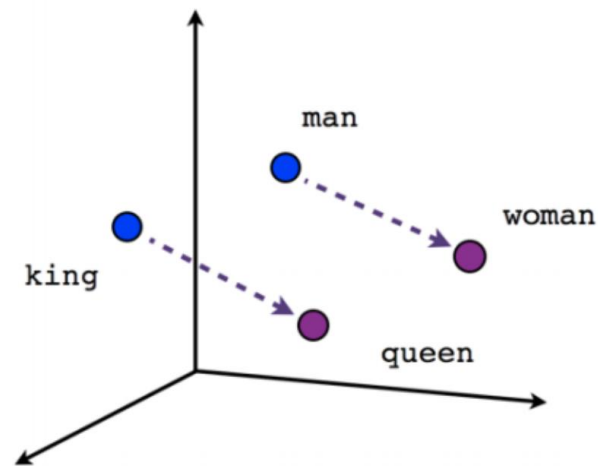
ex) 지점 A에서 지점 B까지의 거리가 어떻게 되나요? → (거리) NUMERIC

ex) 한국의 수도는 어디인가요? → (도시) LOCATION

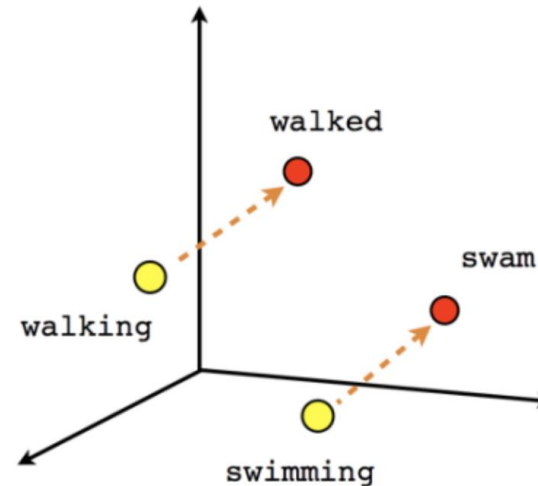
1. Introduction

- 딥 러닝 모델은 컴퓨터 비전과 음성 인식에서 뛰어난 성과를 보였다.
- 자연어 처리(NLP)에서는 단어 벡터 표현 학습과 분류를 위한 벡터 합성에 대한 연구가 주로 이루어지고 있다.
- 단어 벡터는 원핫인코딩 등을 통해 저차원공간으로 투영되며, 투영된 차원에서 의미적 특징을 인코딩하는 특징 추출기이다.
- dense representations에서는 의미론적으로 가까운 단어들은 유클리드나 코사인 거리의 저차원 벡터 공간에서도 가깝게 위치한다.
- 컴퓨터 비전을 위해 발명된 CNN 모델은 NLP에도 효과적이라고 밝혀졌다.
- 특히 의미론적 파싱, 검색 쿼리 검색, 문장 모델링 및 기타 전통적인 NLP 작업에서 우수한 결과를 얻었습니다.

- one-hot encoding은 0과1로 이루어져 행렬의 값이 대부분이 0인 sparse representation(희소표현)으로 공간 낭비를 일으킨다.
- Dense Representation(밀집표현)되는 word embedding인 word2vec을 사용해 각 차원의 값이 실수값을 가지는 벡터로 표현한다.
- 각 단어는 고정된 차원을 갖게 되며, Distributed representation(분산표현)되고, 벡터간 유사도 계산을 할 수 있게 도와준다.
- Word2vec은 특히 단어 위치로 학습을 시켜서 단어를 벡터 공간에 직관적으로 매핑하여, 특정 방향들이 의미나 문맥을 보존한다.



Male-Female



Verb tense

1. Introduction

- 이 연구에서는 비지도 학습 신경망 언어 모델로부터 얻은 단어 벡터 위에 단일 컨볼루션 레이어를 사용하여 간단한 CNN 모델을 훈련한다.
- 이 벡터는 Google News의 1,000억 단어에서 훈련되었으며 공개적으로 이용할 수 있다.
- hyperparameter의 조정이 거의 없음에도 불구하고 간단한 모델로 우수한 결과를 보여준다.
- 이는 미리 학습된 벡터는 'universal' 특성 추출기로서 다양한 분류 task에 활용될 수 있다는 것을 시사한다.
- fine-tuning을 통해 작업 특정 벡터를 학습하면 더욱 향상된 결과를 얻을 수 있다.

2. model

- Input → Convolution → Max pooling → Fully connected layer

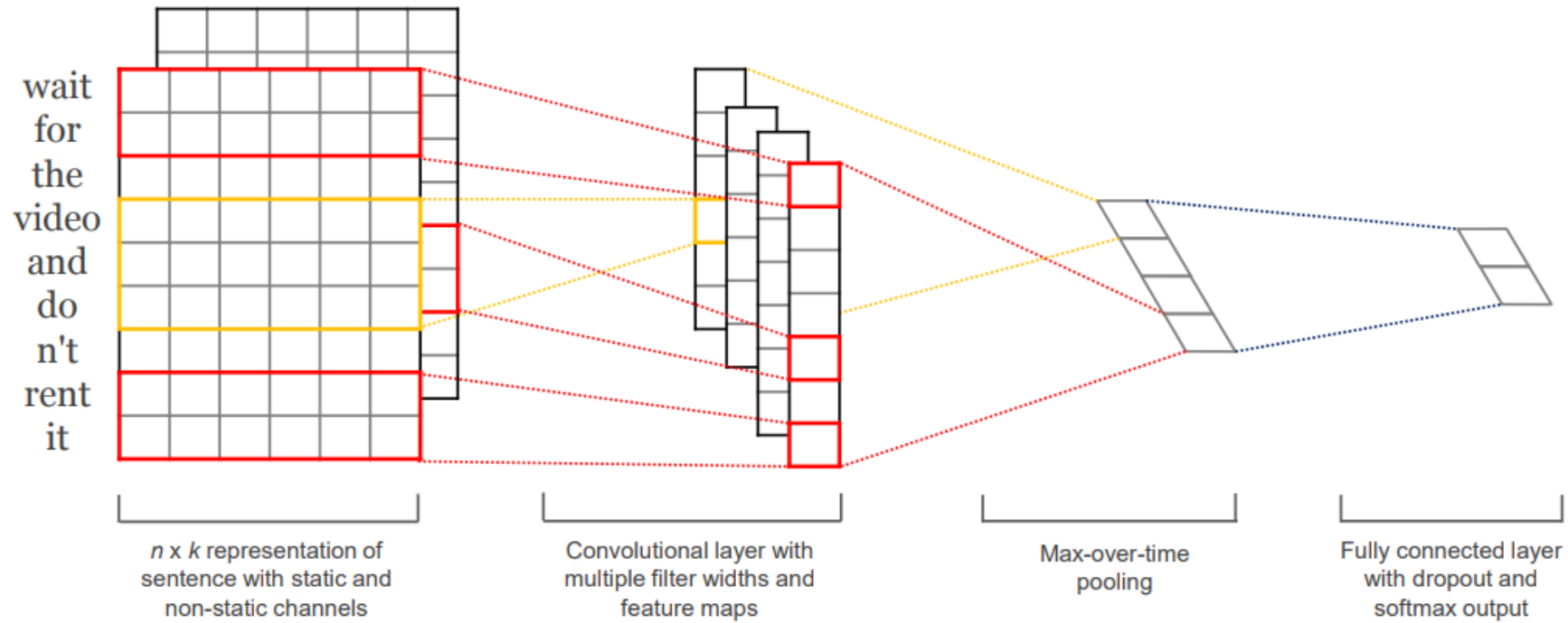
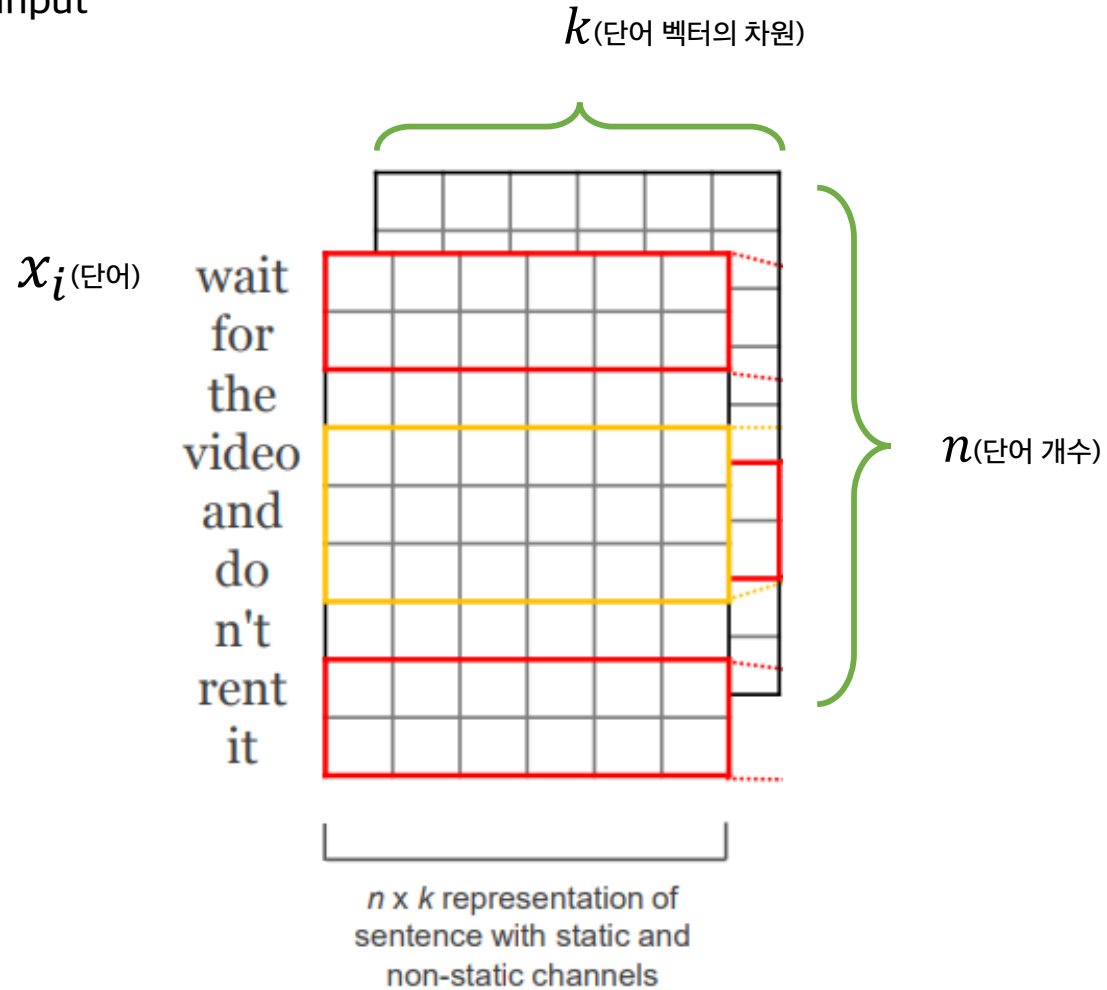


Figure 1: Model architecture with two channels for an example sentence.

2. model

- Input



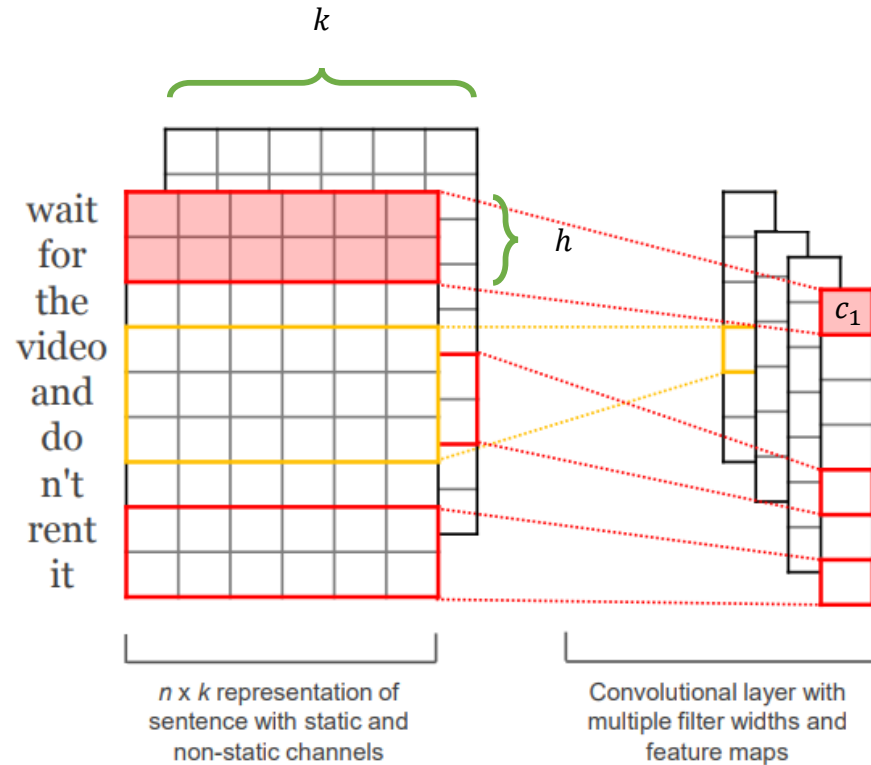
$$\mathbf{x}_{1:n} = \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \dots \oplus \mathbf{x}_n, \quad (1)$$

$\oplus$: concatnation

Wait for the video and do not rent it = $x_{1:9}$

2. model

- Convolution filter(window)

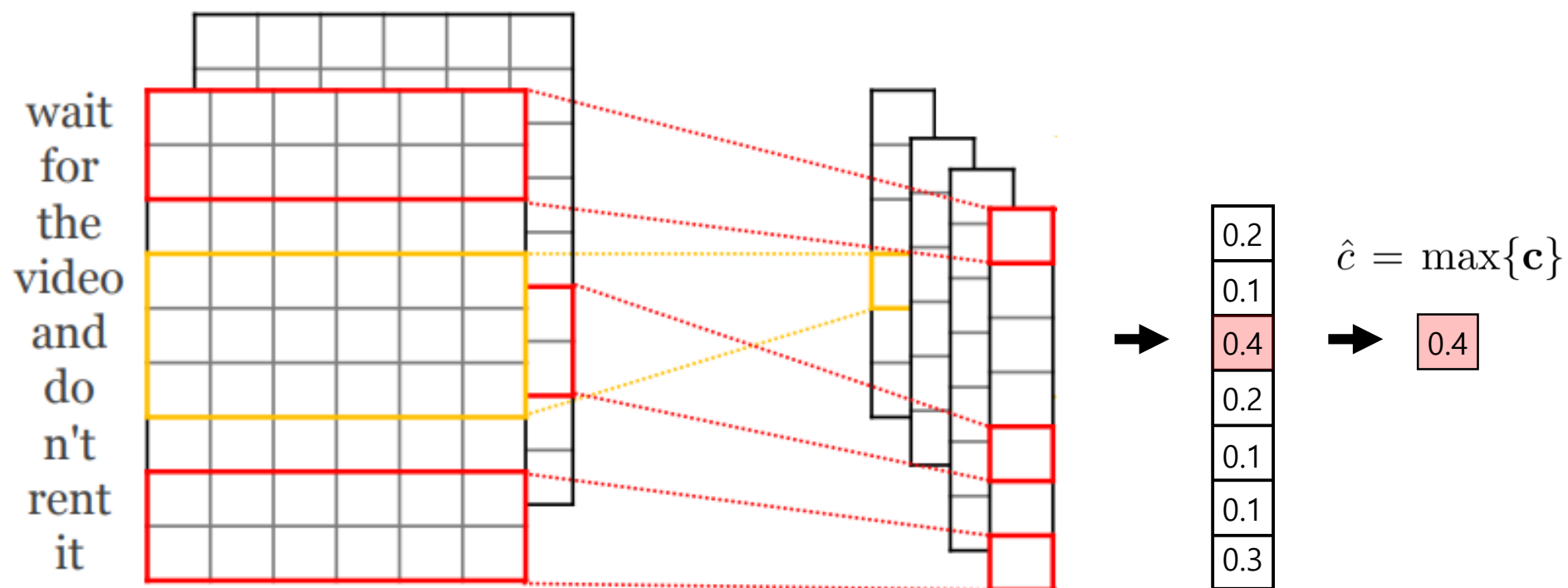


$$c_i = f(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b). \quad (2)$$

- Convolution을 위해 $h * k$ 크기의 filter(window)를 만들어준다
- 단어 벡터의 길이인 k 는 고정
- h 를 조절하여 filter(window) 크기 설정
예) h 가 2일 경우 2개의 단어 씩 특징을 추출하는 것
- 이를 통해 특징 c_i 를 얻음

2. model

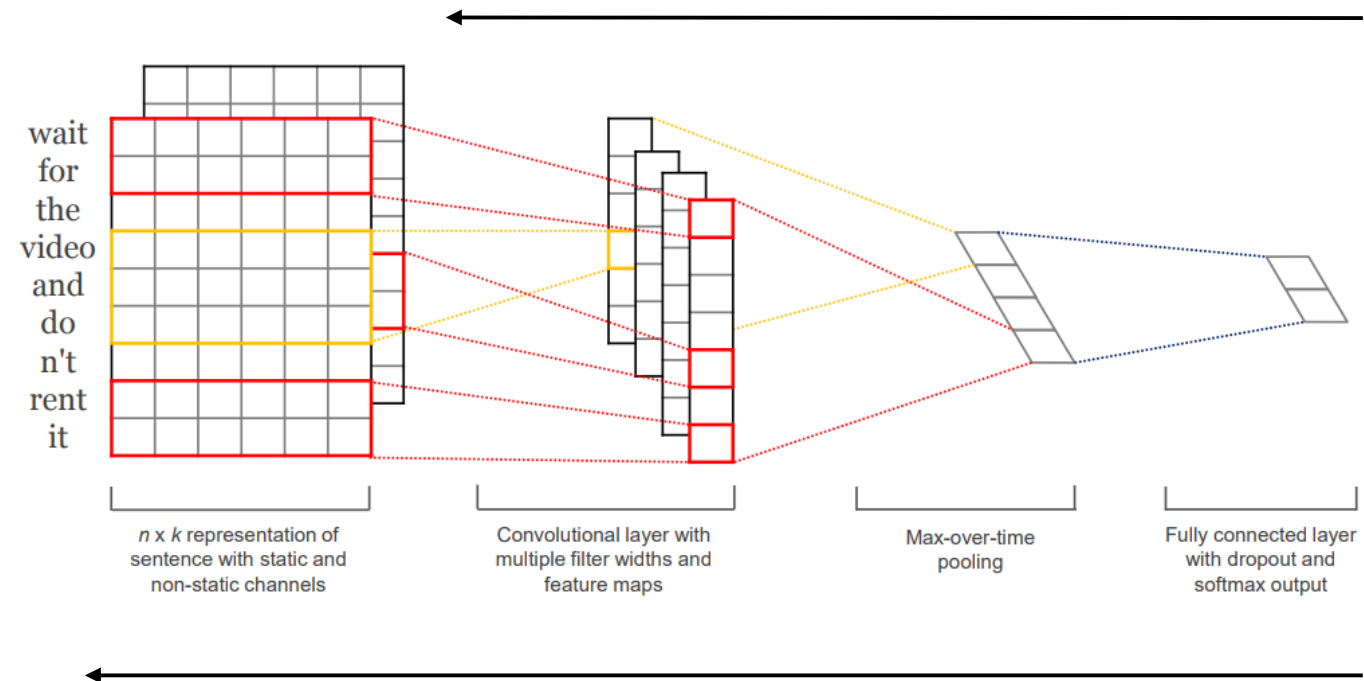
- Max over time pooling(max pooling)



2. model

- static
- non-static

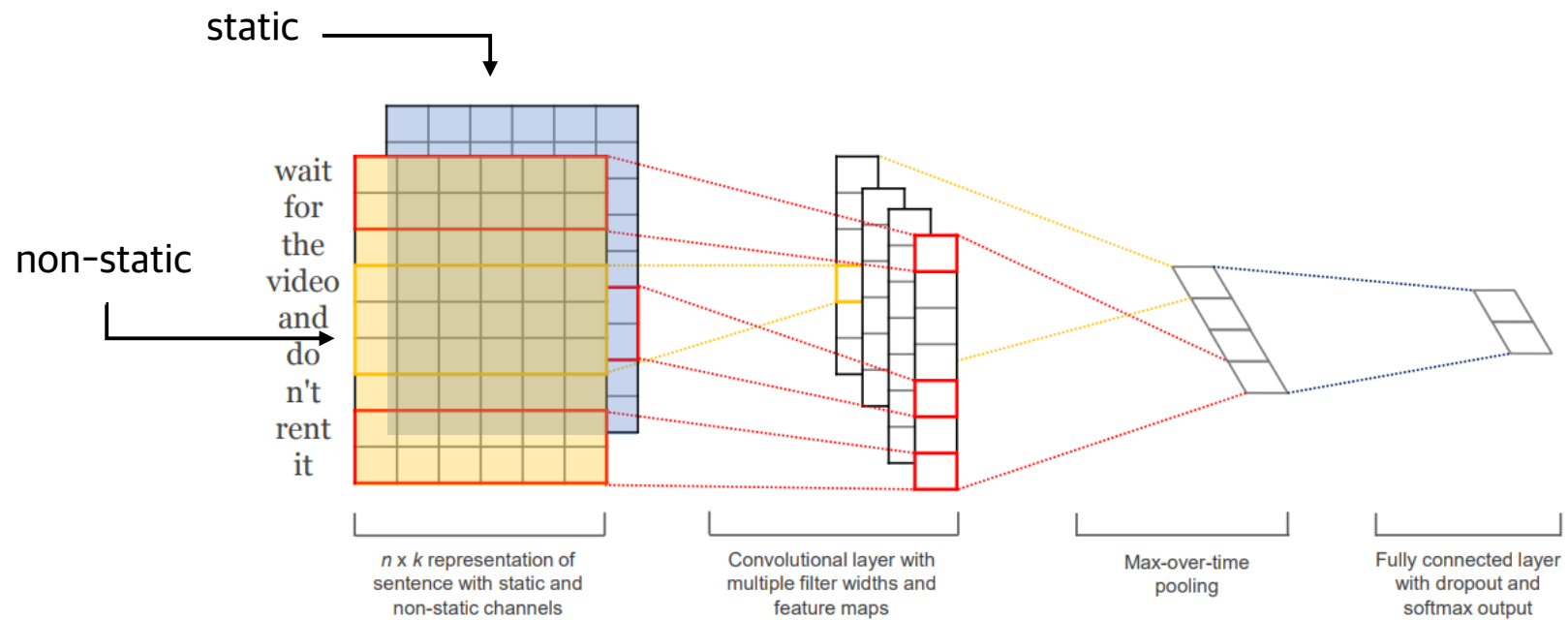
Convolution까지의 학습하여 static이라 부름 (input 단어 벡터를 고정)



Input되는 단어 벡터까지도 학습의 대상으로 포함시켜 이를 non-static이라고 부름

2. model

- Multichannel



3.3 Model Variations

- **CNN-rand**: 기준이 되는 모델, word2vec를 사용하지 않음
- **CNN-static**: word2vec를 사용한 모델
- **CNN-non-static**: CNN-static와 동일하지만, word2vec까지 학습에 사용(fine tuning)한 모델
- **CNN-multichannel**: static, non-static(fine tuning)을 합친 모델로 multi channel을 가짐

3. Datasets and Experimental Setup

- **MR**: 각 리뷰 당 하나의 문장을 가진 영화 리뷰 데이터셋. 긍정/부정에 대한 레이블 제공
- **SST-1**: MR의 확장 버전. 매우 긍정적, 긍정적, 중립적, 부정적, 매우 부정적에 대한 레이블 제공
- **SST-2**: SST-1과 동일하지만 중립적인 리뷰 제거 후 이진 레이블 제공
- **Subj**: 주관성 데이터셋. 문장을 주관적 또는 객관적으로 분류하는 레이블 제공
- **TREC**: TREC 질문 데이터셋으로, 6가지 질문 유형 (사람, 위치, 숫자 정보 등)으로 질문을 분류
- **CR**: 다양한 제품 (카메라, MP3 등)의 고객 리뷰 데이터셋. 긍정적/부정적 리뷰를 예측
- **MPQA** : 짧은 어구에 대한 의견 분류 데이터셋. 긍정, 부정, 중립, 둘 다에 대한 레이블

3.1 Hyperparameters and Training

- ReLU(Rectified Linear Unit)
- filter windows (h) of 3, 4, 5 with 100 feature maps
- dropout rate (p) of 0.5
- l2 constraint (s) of 3
- mini-batch size of 50
- stochastic gradient descent over shuffled mini-batches
- Adadelta update rule

3.3 Model Variations

- 4개의 모델과 기존 모델들과 성능 비교
- 다른 모델들보다 좋은 성능, 7개중 4개가 제일 좋은 성능을 보임

Model	MR	SST-1	SST-2	Subj	TREC	CR	MPQA
CNN-rand	76.1	45.0	82.7	89.6	91.2	79.8	83.4
CNN-static	81.0	45.5	86.8	93.0	92.8	84.7	89.6
CNN-non-static	81.5	48.0	87.2	93.4	93.6	84.3	89.5
CNN-multichannel	81.1	47.4	88.1	93.2	92.2	85.0	89.4
RAE (Socher et al., 2011)	77.7	43.2	82.4	—	—	—	86.4
MV-RNN (Socher et al., 2012)	79.0	44.4	82.9	—	—	—	—
RNTN (Socher et al., 2013)	—	45.7	85.4	—	—	—	—
DCNN (Kalchbrenner et al., 2014)	—	48.5	86.8	—	93.0	—	—
Paragraph-Vec (Le and Mikolov, 2014)	—	48.7	87.8	—	—	—	—
CCAE (Hermann and Blunsom, 2013)	77.8	—	—	—	—	—	87.2
Sent-Parser (Dong et al., 2014)	79.5	—	—	—	—	—	86.3
NBSVM (Wang and Manning, 2012)	79.4	—	—	93.2	—	81.8	86.3
MNB (Wang and Manning, 2012)	79.0	—	—	93.6	—	80.0	86.3
G-Dropout (Wang and Manning, 2013)	79.0	—	—	93.4	—	82.1	86.1
F-Dropout (Wang and Manning, 2013)	79.1	—	—	93.6	—	81.9	86.3
Tree-CRF (Nakagawa et al., 2010)	77.3	—	—	—	—	81.4	86.1
CRF-PR (Yang and Cardie, 2014)	—	—	—	—	—	82.7	—
SVM _S (Silva et al., 2011)	—	—	—	—	95.0	—	—

3.3 Results and Discussion

- 기준 모델(CNN-rand)은 그 자체로는 성능이 좋지 않다
- 사전 훈련된 벡터를 사용하였을 때 성능 향상을 예상했지만, 그 규모에 놀랐다.
- 이러한 결과는 사전 훈련된 벡터가 좋은 'universal' 특징추출기이며, 다양한 데이터셋에서 활용될 수 있다는 것을 말한다.
- 그리고 non-static 모델이 가장 좋은 성능을 보였기 때문에 fine-tuning이 중요한 요인으로 작용된다고 봄

Model	MR	SST-1	SST-2	Subj	TREC	CR	MPQA
CNN-rand	76.1	45.0	82.7	89.6	91.2	79.8	83.4
CNN-static	81.0	45.5	86.8	93.0	92.8	84.7	89.6
CNN-non-static	81.5	48.0	87.2	93.4	93.6	84.3	89.5
CNN-multichannel	81.1	47.4	88.1	93.2	92.2	85.0	89.4

4.1 Multichannel vs. Single Channel Models

- Multichannel이 과적합을 방지하고 더 잘 작동할 것으로 기대했었다.(사전 학습된 벡터가 원래 값에서 너무 멀리 벗어나지 않게 해서)
- fine-tuning process를 규제하는 추가 연구가 필요하다.

예를 들어, non-static 부분에서 대해 Multichannel 대신, 훈련 중에 수정할 수 있는 추가 차원을 사용하는 것으로 single channel을 유지하는 것.

4.2 Static vs. Non-static Representations

	Most Similar Words for	
	Static Channel	Non-static Channel
bad	<i>good</i> <i>terrible</i> <i>horrible</i> <i>lousy</i>	<i>terrible</i> <i>horrible</i> <i>lousy</i> <i>stupid</i>
good	<i>great</i> <i>bad</i> <i>terrific</i> <i>decent</i>	<i>nice</i> <i>decent</i> <i>solid</i> <i>terrific</i>
n't	<i>os</i> <i>ca</i> <i>ireland</i> <i>wo</i>	<i>not</i> <i>never</i> <i>nothing</i> <i>neither</i>
!	<i>2,500</i> <i>entire</i> <i>jez</i> <i>changer</i>	<i>2,500</i> <i>lush</i> <i>beautiful</i> <i>terrific</i>
,	<i>decasia</i> <i>abysmally</i> <i>demise</i> <i>valiant</i>	<i>but</i> <i>dragon</i> <i>a</i> <i>and</i>

- SST-2 데이터셋에서 코사인 유사도를 기반으로 한 가장 가까운 단어 상위 4개
- static 모델에서는 bad에 대해 good이 아마도 구문적으로 동등하다고 생각되고 있다.
- fine-tuning 된 non-static 모델에서는 의미적으로 유사한 단어들로 구성
- fine-tuning 에 포함되지 않는 token의 경우, fine-tuning 을 통해 더 의미 있는 표현을 학습할 수 있다.

4.3 Further Observation

- Kalchbrenner 등(2014)는 CNN-rand와 본질적으로 동일한 아키텍처의 CNN을 사용했지만, SST-1 데이터셋에서 37.4%를 얻었다.
- CNN-rand은 45.0%의 결과를 얻었는데, 이는 더 많은 필터크기와 특성맵을 가지고 있기 때문이라고 설명된다.

Model	MR	SST-1	SST-2	Subj	TREC	CR	MPQA
CNN-rand	76.1	45.0	82.7	89.6	91.2	79.8	83.4

- Dropout이 매우 효과적이었다. Dropout은 일관되게 성능을 상대적으로 2%에서 4% 향상시켰다.
- Word2vec에 없는 단어는 Word2vec와 동일한 분산을 가지도록 샘플링 함으로써 약간의 성능 향상을 얻었습니다.
- Adadelat는 Adagrad와 유사한 결과를 주지만 epochs이 더 적게 듭.

5. Conclusion

- 이 연구는 word2vec 기반 CNN 실험이다
- hyperparameters의 조정이 적고, 간단한 CNN으로 놀랄 만한 성과를 얻었다.
- 이를 통해 단어 벡터의 unsupervised pre-training이 자연어 처리에서 중요한 구성 요소임을 입증한다.

감사합니다.